2000P12740

310

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 926 851 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
30.06.1999 **Bulletin 1999/26**

(51) Int Cl.6: **H04J 3/06**, H04J 3/07, H04J 3/16

(21) Application number: **98310079.3**

(22) Date of filing: **09.12.1998**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **19.12.1997 US 995414**

(71) Applicant: **NORTHERN TELECOM LIMITED**
Montreal, Quebec H2Y 3Y4 (CA)

(72) Inventor: **Bird, Peter Dane**
Kanata, Ontario K2K 1X7 (CA)

(74) Representative:
Humphrey-Evans, Edward John et al
Intellectual Property Law group,
Nortel Networks,
London Road
Harlow, Essex CM17 9NA (GB)

(54) Method of an apparatus for multiplexing and demultiplexing digital signal streams

(57) Multiplexing and demultiplexing are commonplace for an efficient bandwidth utilization in telecommunications. SRTS (Synchronous Residual Time Stamp) technique is widely used for timing recovery in processing of digital signal streams. The bit stuffing is also prevalent for various purposes, one being rate adjustment. The invention performs the SRTS technique entirely digitally to monitor the rate of slower speed signal streams in relation to the rate of a higher speed stream. The digital implementation permits the use of context switching for processing a plurality of digital signal streams. As the result, hardware requirement is greatly reduced.

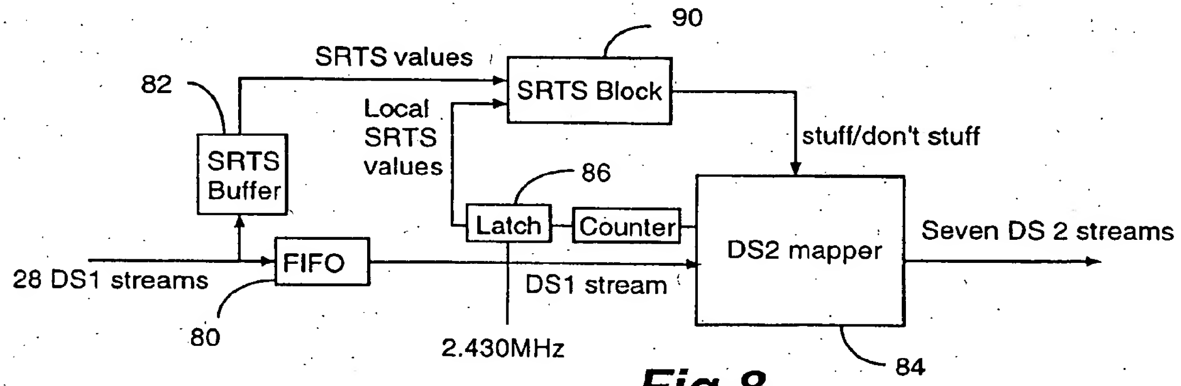


Fig 8

EP 0 926 851 A2

Description

Field of Invention

[0001] The invention resides generally in the field of multiplexing and/or demultiplexing of a plurality of digital signal streams. In particular, it relates to rate adjustment of digital signal streams in connection with multiplexing/ demultiplexing by using SRTS (Synchronous Residual Time Stamp) values.

Background of Invention

[0002] In digital telecommunications systems, digital signal streams are transported at a variety of rates, synchronously or asynchronously, and in all cases, clocks are used for processing these digital signal streams. These clocks must be brought into synchrony for proper data exchange and timing information of some form is transported from the source to destination. Digital signal streams are also frequently regenerated at a proper rate as they experience signal deterioration by noise, delays, jitter etc. They are also multiplexed to produce higher bit rates for higher capacity transmission systems, as a means of utilizing the same transmission medium economically for may different users. In multiplexing, there are more than one digital transmission hierarchies for telecommunications for different parts of the world. Figure 1 shows one example of digital hierarchies used in North America. In the Figure, four DS1 signal streams are multiplexed into one DS2 signal stream and 7 DS2 signal streams are then multiplexed into a DS3 signal stream. The bit rates of levels are also shown in the Figure. A major problem of multiplexing involves synchronizing the input signal streams. Digital signal streams cannot be directly interleaved and guaranteed their subsequent identification unless the pulse rates of all the signal sources are locked to a common clock. This means that all the signal streams are frequency synchronous, or else they are asynchronous and brought into frequency synchronism by the multiplexer. The multiplexer system can establish frequency synchronism by increasing the bit rate of each separate input to one common rate before combining them into one bit stream. This adds extra time slots into which extra pulses can be stuffed. In pulse stuffing, all incoming digital signal streams are stuffed with a sufficient number of pulses to raise each of their rates to that of the desired common rate. In demultiplexing, a single signal stream of a high rate is split into a plurality of signal streams. Each of the split signal streams is stripped of stuffed bits which are so identified and is desynchronized into a single signal stream of a lower rate according to the common clock.

[0003] By referring to Figure 2, this rate adjustment can be explained in more detail. For example, Figure 2 shows the structure of a DS2 signal stream which consists of 1176 bits in a frame called M-frame which is made up of 4 subframes, each of 294 bits. Each subframe includes a stuff block in which one specific bit position is reserved for stuffing. This position is used for stuffing a null bit if stuffing is required or it can be used for data bit if no stuffing is required. Therefore, when the stuff position is used for data, there are 6x48=288 bits of DS1 data in a DS2 M frame. When a null bit is put in the stuff position, there are 287 bits of DS1 data per DS2 frame. Therefore, the DS1 data rate can be adjusted by changing how often the stuff position is used for null bit or data bit. The range of adjustment therefore is from 6.312x287/1176=1.5404 Mb/s to 6.312x288/1176=1.5458 Mb/s.

[0004] Figure 3 shows the structure of DS3 signal stream. A DS3 stream consists 4,760 in an M-frame which is made up of 7 subframes, each of 680 bits. The stuff position in each subframe is also arranged in a similar fashion as in DS2 stream. Therefore in a DS3 stream, there are 7 stuff positions in one M-frame, one per DS2.

[0005] It should be noted that although DS streams are described in detail above, essentially similar digital signal hierarchies are used in Europe and elsewhere such as E1, E2, and E3. The number of bits in a frame, specific bit assignments etc. may be different among them but the concept of the invention is equally applicable to any of these digital signal streams.

[0006] A broadband network such as SONET handles ATM traffic in addition to other synchronous traffic. While a network is synchronous in operation, ATM traffic is asynchronous in nature and thus such a data stream experiences the cell jitter, cell delay variation and other fluctuations as it moves through the network toward a destination node. Even when an ATM network is transporting a CBR (constant bit rate) stream between a source node and a destination node, the clock frequency at the destination node cannot be traced directly back to that of the source node by the regular, periodic arrival of the CBR stream. A few techniques have been used to convey the difference between the network clock frequency and the service clock frequency (frequency of the data being transported). U.S. Patent No. 5,260,978 Nov. 9, 1993 Fleischer et al describes a technique called SRTS (Synchronous Residual Time Stamp) for timing recovery in a broadband network. The patent shows that only small overhead bits (called P bits) are required to express unambiguously the variation from a nominal value of the difference between the two clock frequencies. The P bits are called SRTS bits. The resolution is governed by the nominal value of the frequency difference and the clock tolerance, and determines the required minimum number of P-bits. Under normally expected parameters, these P-bits would be significantly smaller than the number of bits that would be required to express the clock frequencies, e.g. 3 bits instead of 13 bits. A typical sampling period is 3008 which corresponds to a period of 8 ATM cells and a 47 octet payload per cell.

[0007] The SRTS method described in the above-referenced patent and the article makes use of phase locked loops (PLLs) to generate an analog clock signal derived from the SRTS values. U. S. Patent No. 5,608,731 Mar. 4, 1997 Upp et al, on the other hand, describes a digital closed loop clock recovery for SRTS. In the patent, 4 bits are used for SRTS to recover clock pulses of a DS1 signal stream.

[0008] Figure 4 illustrates a functional block diagram in the Tx direction of a typical DS1/DS2 multiplexer (often called M12 multiplexer). As mentioned earlier, in the multiplexer, before combining the lower rate signal streams, the service clock of each lower rate signal stream must be recovered in relation to the network reference clock and then each lower rate signal must be rate adjusted to be synchronous with each other by appropriate stuffing. Therefore in Figure 4, ATM cells from the network are buffered at buffer 30 and a DS1 signal stream is fed to a FIFO 32. Meanwhile, SRTS values are captured from the ATM cell stream. They are then processed at 34 by referring to the network reference clock, e.g., 2.430MHz and a DS1 clock is regenerated through PLL 36. At FIFO, the incoming DS1 signal stream is written in in response to the regenerated DS1 clock. On the other hand, DS1 signal stream stored in the FIFO are read out and sent to DS2 mapper 38 in accordance with the clock signal obtained from DS2 clock 40. Whenever the FIFO reaches a certain threshold of fullness, the DS2 mapper will stuff in DS1 data into the stuff position. When this happens the rate at which the FIFO is emptied increases and eventually the fullness of the FIFO will drop below a threshold 42 and the DS2 mapper will then put nulls into the stuff position. Having adjusted each DS1 signal stream into alignment with each other by proper stuffing, DS2 mapper maps four DS1 signal streams into a DS2 signal stream.

[0009] Figure 5 illustrates a functional block diagram in the Rx direction of a known M12 multiplexer. A DS2 signal stream 50 is disassembled at a disassembler 52 to DS1 signal streams, each DS1 signal stream properly destuffed. Each DS1 signal stream is sent and stored in FIFO 54 according to write clock derived from DS2 clock 56. Each DS1 stream is read out from the FIFO according to a regenerated clock for the respective DS1 stream. The respective clock is regenerated from the data bits coming out of the disassembler and the stuff/don't stuff signal through a PLL 58. The data is read out of the FIFO in accordance with this recovered clock and stored in the ATM cell buffer 60. SRTS values are calculated at 62 with respect to the network reference clock and inserted into the appropriate places in the ATM cells as they are sent to the ATM network.

[0010] While an M12 multiplexer is described above, in actual practice it is more common that DS1 signal streams are multiplexed into a DS3 signal stream at the same location. As shown in Figure 6, this is performed by a series of M12 and M23 multiplexing. In the Figure, seven M12, each with an appropriate stuffing operation at 66 are combined to generate a DS3 signal stream. Some stuffing determination based on various means is performed here.

[0011] When 28 DS1 signal streams are multiplexed to or demultiplexed from a DS3 signal stream, the range of difference in DS1 clock frequencies among all DS1 signal streams is greater than it would be when only one DS1 signal stream is regenerated as done in the above prior art SRTS techniques, producing wider swings in SRTS values.

[0012] The SRTS clock recovery techniques discussed above deal mainly regenerating a digital signal e.g. DS1 signal which is already fairly narrowly bounded. These techniques therefore do not have a sufficient dynamic range to deal with multiplexing and demultiplexing of DS1 and DS3 streams.

[0013] The multiplexer/demultiplexer of the invention makes use of the SRTS technique but it is entirely digital in operation. It also handles digital signal streams of wider hierarchies, such as signals spanning between DS1 and DS3. As the operation is entirely digital, the technique of context switching can also be applied to it. In one embodiment, the contents of the various registers that make up the inventive algorithm are stored in a memory for each of the 28 DS1 signal streams. As each DS1 signal stream is processed, the values for the registers are loaded from the memory, the registers are updated appropriately, and the new values stored back into memory. By making the algorithm completely digital, one avoids an analog PLL, the device pins that it needs, as well as any external components that a PLL may require for each of the 28 DS1 signal stream. Furthermore, context switching allows a single set of hardware registers along with a memory to replace 28 sets of the DS1 hardware registers which would be required for the prior art techniques mentioned above. It is therefore possible to achieve significantly higher levels of integration.

[0014] A copending application Serial No. 08/659,395 filed on June 6, 1996, inventors Coady et al describes in detail the invention relating to context switching of a plurality of digital signal streams.

Objects of the Invention

[0015] It is therefore an object of the invention to provide a method of and apparatus for multiplexing/demultiplexing digital signal streams of different hierarchy levels.

[0016] It is a further object of the invention to provide a method of and apparatus for determining the stuffing ratio in response to the measurement of clock frequency.

[0017] It is another object of the invention to provide a method of and apparatus for generating SRTS values.

[0018] It is yet another object of the invention to provide a method of and apparatus for performing stuffing digital signal streams.

[0019] It is still an object of the invention to provide a method of and apparatus for processing a plurality of digital signal streams in which context switching is applied to algorithm executions.

Summary of the Invention

[0020] Briefly stated, according to one aspect, the invention is directed to a multiplexer of a plurality of first signal streams into a second signal stream, the first signal streams having a substantially similar first rate and the second signal stream having a second rate higher than the first rate. The multiplexer includes a buffer for buffering the first signal streams, a SRTS circuit for receiving a SRTS value of each first signal stream and a local SRTS circuit for generating a local SRTS value of each first signal stream based on a reference clock rate. The multiplexer includes further a stuffing circuit for stuffing the plurality of first signal streams in response to a stuff command signal which has been generated for each first signal stream in response to its respective SRTS value and local SRTS value and a combiner circuit for combining all the first signal streams into the second signal stream.

[0021] According to another aspect, the invention is directed to a demultiplexer for demultiplexing a first signal stream having a first rate into a plurality of second signal streams, each having a substantially similar rate that is lower than the first rate. The demultiplexer comprises a disassembling circuit for disassembling the first signal stream into the plurality of second signal streams and an incoming SRTS circuit for generating an incoming SRTS value for each of the second signal streams. The demultiplexer includes further an SRTS block for generating an outgoing SRTS value for each of the second signal streams in response to a difference between the incoming and outgoing SRTS values and an inserting circuit for inserting the outgoing SRTS value in each respective second signal stream.

[0022] According to yet another aspect, the invention resides in a process of multiplexing and demultiplexing between a plurality of first signal streams and a second signal stream, the first signal streams having a substantially similar first rate and the second signal stream having a second rate higher than the first rate. The invention is directed to a method of monitoring the first rate relative to the second rate which comprises steps of receiving a SRTS value of each first signal stream, generating a local SRTS value based on the second rate of the second signal stream, and calculating from the received SRTS value and the generated local SRTS value a difference indication proportional to the difference between the first and the second rates.

Brief Description of the Drawings

[0023] Figure 1 is a chart showing a digital hierarchy used in the North America.

[0024] Figure 2 shows the DS2 frame construction.

[0025] Figure 3 shows the DS3 frame construction.

[0026] Figure 4 is a brief functional block diagram of a M12 multiplexer of a know design in the Tx direction.

[0027] Figure 5 is a brief functional block diagram of a M12 multiplexer of a know design in the Rx direction.

[0028] Figure 6 shows a known M13 multiplexer which is a combination of M12 and M23

[0029] Figure 7 is a functional block diagram multiplexer/demultiplexer which makes use of context switching according to one embodiment of the invention.

[0030] Figure 8 is a functional block diagram of M12 in the Tx direction, according to one embodiment of the invention which includes a mechanism for generating a stuff command signal.

[0031] Figure 9 is a functional block diagram of M12 in the Rx direction according to one embodiment of the invention which includes a mechanism for generating SRTS values.

[0032] Figure 10 is a detailed block diagram of SRTS block 80 shown in Figure 8.

[0033] Figure 11 is a detailed block diagram of SRTS block 90 shown in Figure 9.

Detailed Description of the Preferred Embodiments of the Invention

[0034] Figure 7 is a functional block diagram of DS1-DS3 multiplexer/demultiplexer according to one embodiment of the invention. In this Figure, DS1 or DS0 signal streams are received from or sent to an ATM network or some other network and multiplexed into or demultiplexed from a DS3 signal stream. An interface arrangement 70 produces an octet stream of 28 DS1 signal streams in the Tx direction and receive them in the Rx direction. Each of 28 DS1 signal streams is processed by M12 mux which contains various sets of algorithms 72 and 74, e.g., a framing algorithm, SRTS algorithm, stuffing algorithm etc. The SRTS and stuffing algorithms will be described in detail below. These algorithms are executed on each of 28 DS1 signal streams by context switching in co-operation with a memory 76 and therefore the output of M12 is seven DS 2 signal streams. Each of seven DS2 signal streams is processed by multiplexer M23 which, like M12, contains necessary algorithms and executed by context switching. Seven DS2 signal streams are then combined and mapped onto a properly framed DS3 signal stream. In the Rx direction, a DS3 signal stream is processed by similar algorithms at M23 and M12. At M12, rate adjustment is performed by SRTS and stuffing algorithms, which will be described in detail below also.

[0035] As mentioned earlier, the process of multiplexing up to a DS3 and demultiplexing down from a DS3 introduces a much greater variability in the SRTS period than is seen in a normal period of 3008 DS1 bits as assumed by the prior art. This variability is due to a number of factors:

- jitter on the DS3 signal stream,

- the frame format of the DS3 which adds Frame bits, and stuff opportunities at the DS2 level,
- jitter on the DS2 signal stream which can lead to saturation of the DS2 to DS3 stuffing mechanism which can lead to extended periods of all null stuffing and all data stuffing,
- the frame format of the DS2 which adds Frame bits and stuff opportunities at the DS1 level,
- jitter on the DS1 signal stream which can lead to saturation of the DS1 to DS2 stuffing mechanism which can lead to extended periods of all null stuffing and all data stuffing.

[0036] The overall effect of these factors is that the total elapsed time in which 3008 DS1 bits arrive via the DS3 signal stream or are sent out via the DS3 signal stream exceeds the bounds set by the 4 bit SRTS value which is used in the above-noted prior art.

[0037] Figure 8 is a functional block diagram of a multiplexer of DS1 signal streams to a DS2 signal stream, containing an embodiment of the invention. In the Figure, each of a plurality (e.g. 28) of DS1 signal streams is buffered at FIFO 80 and its SRTS value is recovered by a SRTS recovery buffer 82. This sequence of SRTS values is an indication of the rate of an incoming DS1 signal stream. A DS2 mapper 84 reads a DS1 stream stored in the FIFO. Having appropriately rate adjusted by stuffing, the DS2 mapper maps four DS1 signal streams into an outgoing DS2 signal stream. The stuffing action as well as DS2 framing takes place in the mapper 84. Decision to stuff the stuff position with a null bit or data bit is performed at SRTS Block 90. Therefore, a counter reads DS1 bits of a particular DS1 signal stream which is mapped to the DS2 signal stream. The counter performs a division by 3008 and latches the clock count of a network reference clock, e.g., 2.430MHz. The content of the latch is the local SRTS value which is an indication of the rate of the particular outgoing DS1 stream multiplexed into the DS2 stream. The SRTS Block 90 compares the sequence of the recovered SRTS values and the sequence of local SRTS values and generates a command of stuff null or stuff data at a stuff position with respect to each of the 28 DS1 signal streams.

[0038] According to one embodiment, as in the case of multiplexing DS1 signal streams to a DS2 signal stream, each DS2 signal stream is further stuffed while a DS3 mapper maps seven DS2 signal streams into a DS3 signal stream. Seven DS2 signal streams are individually stuffed to align all the DS2 signal streams.

[0039] Figure 9 is a functional block diagram of a demultiplexer of a DS2 signal stream to DS1 signal streams, containing an embodiment of the invention. In the Figure each of seven DS2 signal streams is demultiplexed into 4 DS1 signal streams at disassembler 92. Each of a plurality of DS1 signal streams is buffered in a FIFO 94. A counter 96 counts the DS1 bits for a particular DS1 signal stream taking into account the presence or absence of a data bit in each stuff opportunity. The counter divides by 3008 and latches at latch 98 the clock count of a network reference clock, e.g., 2.430MHz. The content of the latch is the incoming SRTS value, the sequence of which is an indication of the rate of the particular incoming DS1 stream. The SRTS block 100 generates an outgoing SRTS value, the sequence of which is an indication of the rate of a particular outgoing DS1 stream. The SRTS block compares the incoming SRTS sequence and the outgoing SRTS sequence and alters the outgoing SRTS sequence so that the outgoing rate for a particular DS1 signal stream will match the incoming rate for that DS1 signal stream. This SRTS sequence is embedded in each respective DS1 stream.

[0040] The function of SRTS block 90 of Figure 8 consists of two parts. The first is used to define a stuffing ratio required to align the incoming data rate as defined by the sequence of incoming SRTS values and the outgoing data rate as defined by the sequence of local SRTS values. The second part produces a sequence of stuff values corresponding to the stuffing ratio determined above, which controls the outgoing data rate and feeds this back to the first section.

Part 1 - data rate alignment

[0041] The transmit DS3 channel is driven by a TX clock rate and it is necessary to provide a data bit for each TX clock. Each data bit may correspond to a DS3 frame bit or one of the seven DS2 channels (for which it may be a data bit or a stuff bit). In turn, each bit for a DS2 channel may correspond to a frame bit or one of the 28 DS1 channels where it too may be a data bit or a stuff bit.

[0042] The process for data rate alignment is triggered by the request to send DS1 bits. When a particular DS1 has requested a total of 3008 bits, a new local SRTS value is computed and a new incoming SRTS value is read. The difference between successive local SRTS values is computed and is an indication of the clock rate for the outgoing DS1 bits. The difference between incoming SRTS values is calculated and is an indication of the clock rate of the incoming DS1 data stream. The difference between these differences is a measure of how far apart the incoming and outgoing clock rates are. The circuit then accumulates the difference of differences and attempts to adjust the outgoing SRTS rate so that the cumulative difference is zero thereby ensuring that the two frequencies are in alignment. In this process, the local delta SRTS values will be varying considerably, whereas the outgoing delta SRTS values will be smoothed to a roughly constant value.

Part 2 - stuffing ratio tracking:

[0043] For a given DS2 rate, nominally 6.312Mb/s, the outgoing DS1 data rate can be controlled by altering the stuffing ratio that is used. Referring back to Figure 4, within a DS2 sub-frame either 287 or 288 bits of DS1 data can be transmitted. This means that on average, over a large number of frames, the range of DS1 rates that can be supported is between 287/1176*6.312=1.5404 Mb/s and 288/1176*6.312=1.5458 Mb/s (there are 1176 bits per DS2 frame).

[0044] Part one of the invention above determines the data rate of the outgoing data stream. This data rate is expressed as a desired stuffing ratio that will range between 0 and 100%. Whatever the stuffing ratio is, part two of the invention produces a sequence of stuff/don't stuff signals that will correspond to the desired stuffing ratio.

[0045] For example, a nominal DS1 data rate of 1.544Mb/s is to be produced, this corresponds to a stuffing ratio of 66% which in turn translates to 287.665 (=1.544/6.312*1176) data bits per sub-frame. The correct sequence of stuff/don't stuff signals for this data rate is then:

110110110... where 1 means that a data bit is added and 0 means that a null bit is added.

[0046] In order to generate a sequence of 0s and 1s that average out to the desired stuffing ratio, one represents the stuffing ratio as a ratio between an N bit target value and an N bit denominator, i.e., target/2^N. With this representation, the stuffing ratio can range from 0 to (2^N-1)/ (2^N). If the target value is summed 2^N times, then the total will be (2^N)*target. If the sum is computed modulo 2^N, and a 1 selected whenever the sum is 2^N or greater and 0 whenever the sum is less than 2^N, then after 2^N samples a 1 will be produced exactly target times and a 0 {(2^N)-target} times. The average value will be exactly that required, namely target/(2^N), and the sequence of ones and zeros will be distributed as evenly as possible, thereby minimizing the error of the average stuffing ratio with respect to the desired stuffing ratio.

[0047] Using a fraction to represent the desired stuffing ratio will introduce an inherent error between the actually clock rate and that indicated by the desired rate. This can be minimized by making N sufficiently large, however, the process is self correcting. For a given target value, the outgoing and incoming clock rates may differ which will lead to a drift in the respective SRTS differences. When this difference is sufficient, the target value will be adjusted by one thereby causing the outgoing clock rate to change so that the differences will drift in the opposite direction. This will continue indefinitely with the average of the two target values reflecting the true incoming clock rate.

[0048] The function of SRTS block 100 of Figure 9 consists also of two parts. These functions are almost identical to those used for multiplexer. Like multiplexer, the first is used to define a target delta SRTS value required to align the incoming data rate as defined by the sequence of incoming SRTS values and the outgoing data rate as defined by the sequence of outgoing SRTS values. The second part produces a sequence of SRTS values whose difference corresponds to the target delta SRTS value determined above, which controls the outgoing data rate and feeds this back to the first section.

[0049] Referring to Figures 10 and 11, SRTS logic block 90 of Figure 8 for Tx direction and SRTS logic block 100 of Figure 9 for Rx direction are described below in more detail in connection with one embodiment of the invention.

[0050] Referring to Figure 10 in particular, an incoming SRTS value from the network is 4 bits wide which is converted to 5 bits wide at 110. A DS1 stream (DS1 bit) is divided by 3008 which then causes the 5 bit counter of the network reference clock to be latched at latch 112. The content of the latch is the local SRTS value. Delta local SRTS is a difference between successive local SRTS values and is calculated at delta_srts 114. Meanwhile, delta incoming SRTS is a difference between successive incoming SRTS values and is calculated at delta_srts 116. A difference between two deltas, delta local SRTS and delta incoming SRTS, is accumulated at 118 which produces cumulative delta SRTS, cd_srts_R. The cumulative delta SRTS is then filtered at 120 to help remove the jitter due to the framing formats, DS2 and DS3 jitter in addition to source jitter and SRTS variability. Using the filtered cumulative delta SRTS value, fcd_srts_R, as an error function, this is applied to the proportional/integrating control system implemented by 122 and 124. The result is a stuffing target value that is controlled so that the filtered cumulative delta SRTS value tends to zero. With a stuffing target so generated, the stuffing sequence generator, 126, will produce a sequence of stuff null and stuff data that averages out to the ratio specified by the stuffing target.

[0051] Referring to Figure 11, a DS1 stream (DS1 bit) 140 from the demultiplexer, is divided by 3008 which then causes the 5 bit counter 142 of the network reference clock to be latched at latch 144. The content of the latch represents the incoming SRTS value. A difference between successive incoming SRTS values is computed at delta_srts 146. At the same time, a difference of successive outgoing SRTS values is computed at delta_srts 148 and converted to a 5 bit value. Another difference of these two deltas is computed and accumulated at cd_srts_R 150. This is the cumulative delta SRTS value. This value is then filtered at 152 to remove jitter due to the framing formats, DS2 and DS3 jitter in addition to source jitter and outgoing SRTS variability. Using the filtered cumulative delta SRTS value, fcd_srts_R, as an error function, this is applied to the proportional/ integrating control system implemented by 154 and 156. The result is a delta SRTS target that is controlled so that the filtered cumulative delta SRTS value tends to zero. With a delta SRTS value so generated, the SRTS sequence generator 158 will produce a sequence of SRTS values such that the difference between successive values averages out to the value specified by the target.

[0052] According to yet a further embodiment, referring to Figures 10 and 11, the contents of the various registers that make up the algorithm are stored in a memory for each of the 28 DS1s. As each DS1 is processed, the values for the registers are loaded from the memory, the registers are updated appropriately, and the new values stored back into memory. Furthermore, context switching allows a single set of hardware registers along with a memory can replace 28 sets of the DS1 hardware registers which would be required for the prior art techniques mentioned above. It is therefore possible to achieve much higher levels of integration.

Claims

1. A multiplexer of a plurality of first signal streams into a second signal stream, the first signal streams having a substantially similar first rate and the second signal stream having a second rate higher than the first rate, comprising:

   a buffer for buffering the first signal streams;
   an SRTS circuit for receiving SRTS values of each first signal stream;
   a local SRTS circuit for generating local SRTS values of each first signal stream based on a reference clock rate;
   a stuffing circuit for stuffing the plurality of first signal streams in response to a stuff command signal which has been generated for each first signal stream in response to its respective sequences of SRTS values and local SRTS values; and
   a combiner circuit for combining all the first signal streams into the second signal stream.

2. A multiplexer according to claim 1 further comprising:

   the SRTS circuit including delta circuits for calculating a first difference of successive SRTS values;
   the local SRTS circuit including local delta circuits for calculating a second difference of successive local SRTS values, and
   a cumulative delta circuit for registering a cumulative value of a third difference between the first and second differences for each first signal stream.

3. A multiplexer according to claim 2 further comprising:
   a stuff command circuit for generating the stuff command signal in response to the cumulative value of the third difference.

4. A multiplexer according to any one of claims 1-3 further comprising:

   a memory for storing instantiation values of the SRTS circuit, local SRTS circuit and stuffing circuit for each first signal stream, and
   a processor for performing context switching the executions of the SRTS circuit, local SRTS circuit and the stuffing circuit for each first signal stream.

5. A multiplexer according to claim 3 further comprising:

   a memory for storing instantiation values of the SRTS circuit, local SRTS circuit, delta SRTS circuits, delta local SRTS circuit, stuffing circuit, and stuff command circuit for each first signal stream, and
   a processor for performing context switching the executions of the SRTS circuit, local SRTS circuit, delta SRTS circuits, delta local SRTS circuit, stuffing circuit, and stuff command circuit for each first signal stream.

6. A multiplexer according to claim 3 wherein the stuff command circuit comprises:
   a stuff target circuit for comparing the cumulative value of the third difference and a target stuff ratio to generate the stuff command signal.

7. A multiplexer according to any one of claims 1-6 wherein the first and second signal streams are chosen from any one group of the followings, DS1, DS2, and DS3, and E1, E2 and E3.

8. A demultiplexer for demultiplexing a first signal stream having a first rate into a plurality of second signal streams, each having a substantially similar rate that is lower than the first rate, comprising:

   a disassembling circuit for disassembling the first signal stream into the plurality of second signal streams,
   an incoming SRTS circuit for generating an incoming SRTS value for each of the second signal streams;
   an SRTS block for generating an outgoing SRTS value for each of the second signal streams in response to a difference between the incoming and outgoing SRTS values; and
   an inserting circuit for inserting the outgoing SRTS value in each respective second signal stream.

9. A demultiplexer according to claim 8 further comprising:

   the incoming SRTS circuit including incoming

delta circuits for calculating a fourth difference of successive incoming SRTS values; and
the SRTS block including outgoing delta circuits for calculating a fifth difference of successive outgoing SRTS values and a cumulative delta circuit for registering a cumulative value of a sixth difference between the fourth and fifth differences for each second signal stream.

10. A demultiplexer according to claim 9 further comprising:

a memory for storing instantiation values of the incoming SRTS circuit and SRTS block for each first signal stream, and
a processor for performing context switching the executions of the SRTS circuit, local SRTS circuit and the stuffing circuit for each first signal stream.

11. A demultiplexer according to any one of claims 9-10 wherein the SRTS block comprises:
a stuff target circuit for comparing the cumulative value of the sixth difference and a target SRTS value to generate the outgoing SRTS value.

12. A demultiplexer according to any one of claims 8-11 wherein the first and second signal streams are chosen from any one group of the followings, DS1, DS2, and DS3, and E1, E2 and E3.

13. In a process of multiplexing and demultiplexing between a plurality of first signal streams and a second signal stream, the first signal streams having a substantially similar first rate and the second signal stream having a second rate higher than the first rate, a method of monitoring the first rate relative to the second rate, comprising steps of:

receiving a SRTS value of each first signal stream;
generating a local SRTS value based on the second rate of the second signal stream; and
calculating from the received SRTS value and the generated local SRTS value a difference indication proportional to the difference between the first and the second rates.

14. A method according to claim 13 comprising a further step of:
generating a stuff command signal from the difference indications so that all the first signal streams are in alignment.

15. A method according to any one of claims 13-14 comprising a further step of:
generating a new SRTS value for each first signal stream from the difference indication.

16. A method according to any one of claims 13-15 comprising further steps of:

calculating a first difference between two successive SRTS values for each first signal stream; and
calculating a second difference between two successive local SRTS values.

17. A method according to claim 16 comprising a further step of:
generating a stuff command signal from the difference indications to align all the first signal streams.

18. A method according to any one of claims 16-17 comprising a further step of:
generating a new SRTS value for each first signal stream from the difference indication.

19. A method according to any one of claims 17-18 comprising further steps of:

performing all the steps recited therein separately for each first signal stream, and
context switching execution of said all steps among first signal streams.

20. A method according to any one of claims 16-19 wherein the first and second signal streams are chosen from any one group of the followings, DS1, DS2, and DS3, and E1, E2 and E3 and.

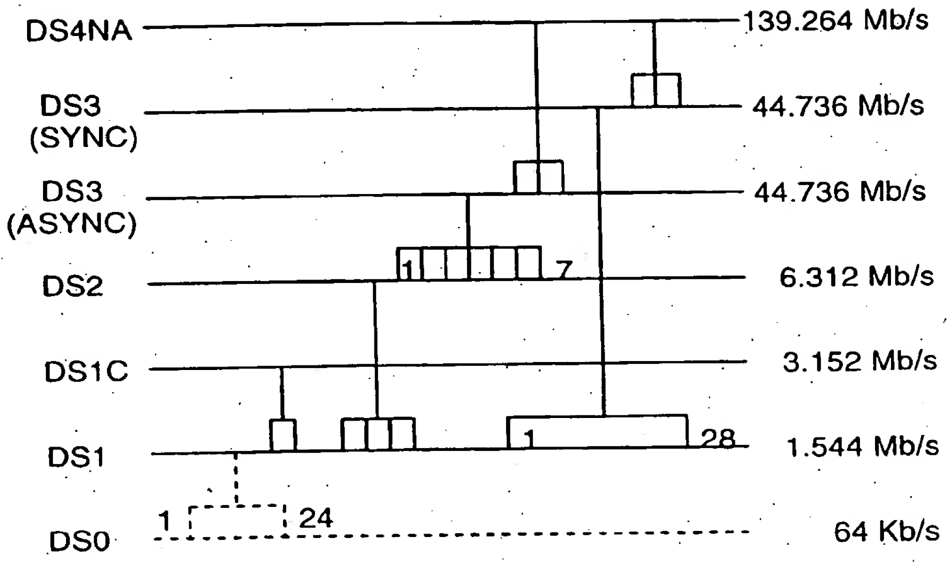
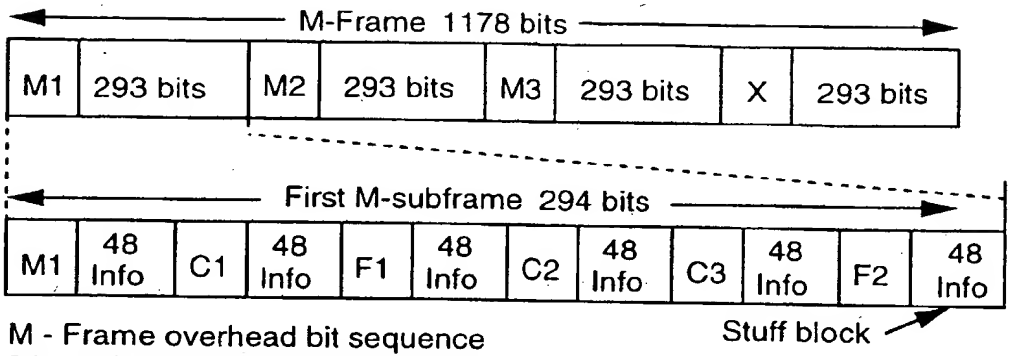
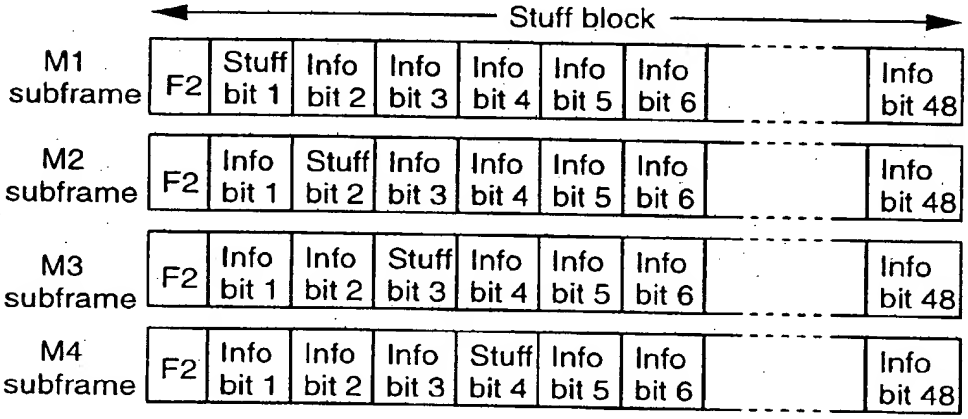

DS4NA
DS3
(SYNC)
DS3
(ASYNC)
DS2
DS1C
DS1
DS0
139.264 Mb/s
44.736 Mb/s
44.736 Mb/s
6.312 Mb/s
3.152 Mb/s
1.544 Mb/s
64 Kb/s
1
7
1
28
1
24

Fig 1


M-Frame 1178 bits
M1
293 bits
M2
293 bits
M3
293 bits
X
293 bits
First M-subframe 294 bits
M1
48 Info
C1
48 Info
F1
48 Info
C2
48 Info
C3
48 Info
F2
48 Info
Stuff block

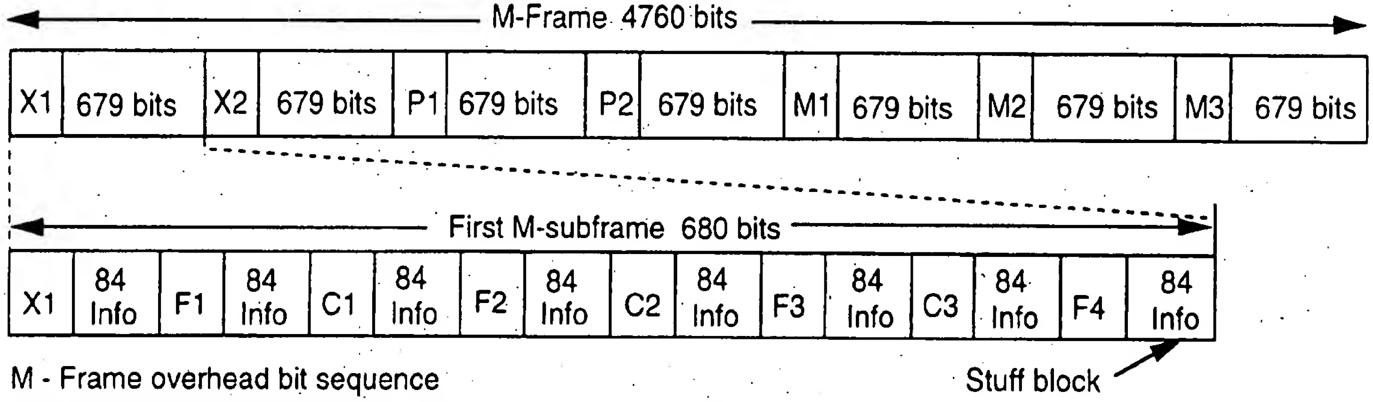
M - Frame overhead bit sequence
24 overhead bits occupies sequenctial overhead bit positions as follows
M1, C1, F1, C2, C3, F2
M2, C1, F1, C2, C3, F2
M3, C1, F1, C2, C3, F2
X, C1, F1, C2, C3, F2


Stuff block
M1 subframe
F2
Stuff bit 1
Info bit 2
Info bit 3
Info bit 4
Info bit 5
Info bit 6
Info bit 48
M2 subframe
F2
Info bit 1
Stuff bit 2
Info bit 3
Info bit 4
Info bit 5
Info bit 6
Info bit 48
M3 subframe
F2
Info bit 1
Info bit 2
Stuff bit 3
Info bit 4
Info bit 5
Info bit 6
Info bit 48
M4 subframe
F2
Info bit 1
Info bit 2
Info bit 3
Stuff bit 4
Info bit 5
Info bit 6
Info bit 48

Notes

1. The M-frame alignment signal is M1=0, M2=1, and M3=1
2. The M-subframe alignment signal is F1=0 and F2=1
3. The C1, C2 and C3 bit positions are available for application specific use
4. In the M12 multiplex application, stuffing for ith DS1 channel occurs in M-subframe i in the ith information bit of the last block. The C-bits of that M-subframe are set to C1=C2=C3=0 indicating no stuffing.

Fig 2


M-Frame 4760 bits
X1
679 bits
X2
679 bits
P1
679 bits
P2
679 bits
M1
679 bits
M2
679 bits
M3
679 bits
First M-subframe 680 bits
X1
84 Info
F1
84 Info
C1
84 Info
F2
84 Info
C2
84 Info
F3
84 Info
C3
84 Info
F4
84 Info
Stuff block

M - Frame overhead bit sequence
55 overhead bits occupies sequenctial overhead bit positions as follows

X1, F1, C1, F2, C2, F3, C3, F4
X1, F1, C1, F2, C2, F3, C3, F4
P1, F1, C1, F2, C2, F3, C3, F4
P2, F1, C1, F2, C2, F3, C3, F4
M1, F1, C1, F2, C2, F3, C3, F4
M2, F1, C1, F2, C2, F3, C3, F4
M3, F1, C1, F2, C2, F3, C3, F4

Fig 3

Notes

1. The M-frame alignment signal is M1=0, M2=1, and M3=0
2. The M-subframe alignment signal is F1=1, F2=0, F3=0, and F4=1
3. The C1, C2 and C3 bit positions are available for application specific use

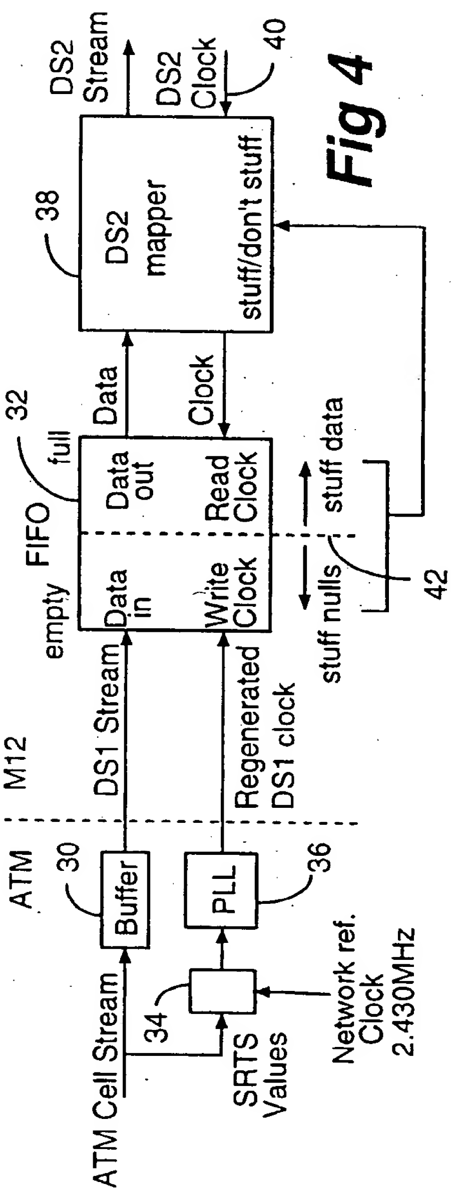

ATM Cell Stream
34
SRTS Values
Network ref. Clock 2.430MHz
Buffer
30
ATM
PLL
36
M12
DS1 Stream
Regenerated DS1 clock
empty
FIFO
full
32
Data in
Write Clock
Data out
Read Clock
stuff nulls
stuff data
42
Data
Clock
38
DS2 mapper
stuff/don't stuff
DS2 Stream
DS2 Clock
40

Fig 4

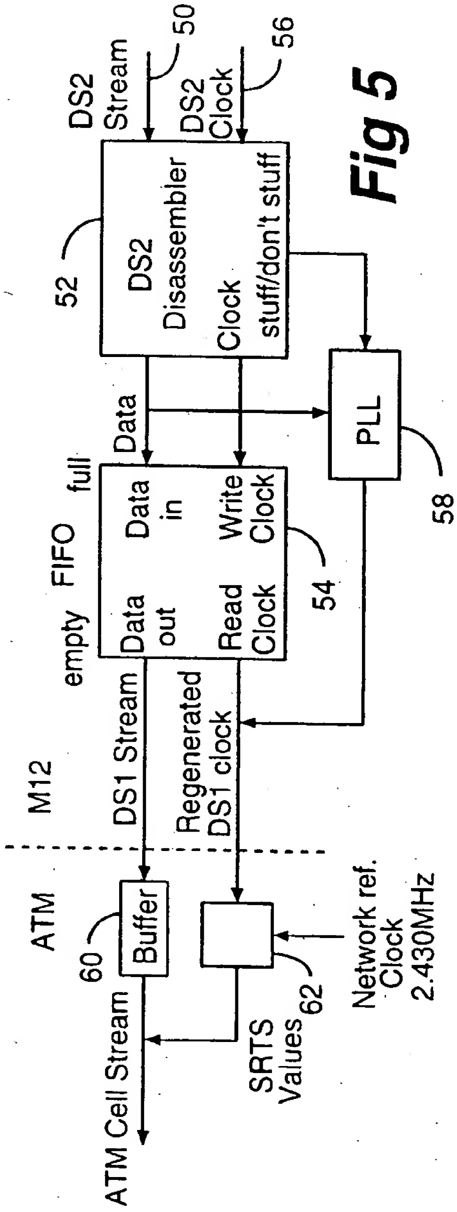

ATM Cell Stream
SRTS Values
62
Network ref. Clock 2.430MHz
60
Buffer
ATM
M12
DS1 Stream
Regenerated DS1 clock
empty
FIFO
full
Data out
Read Clock
Data in
Write Clock
54
Data
PLL
58
52
DS2 Disassembler
Clock
stuff/don't stuff
DS2 Stream
50
DS2 Clock
56

Fig 5

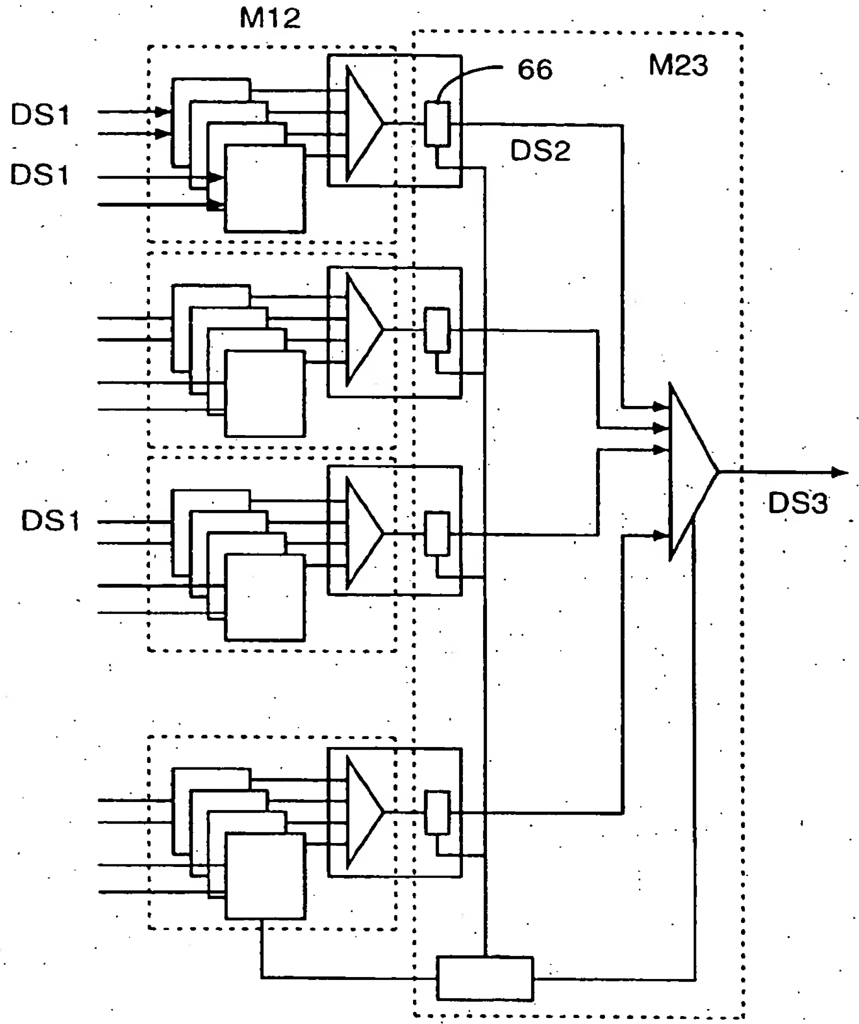

M12
66
M23
DS1
DS1
DS2
DS1
DS3

Fig 6

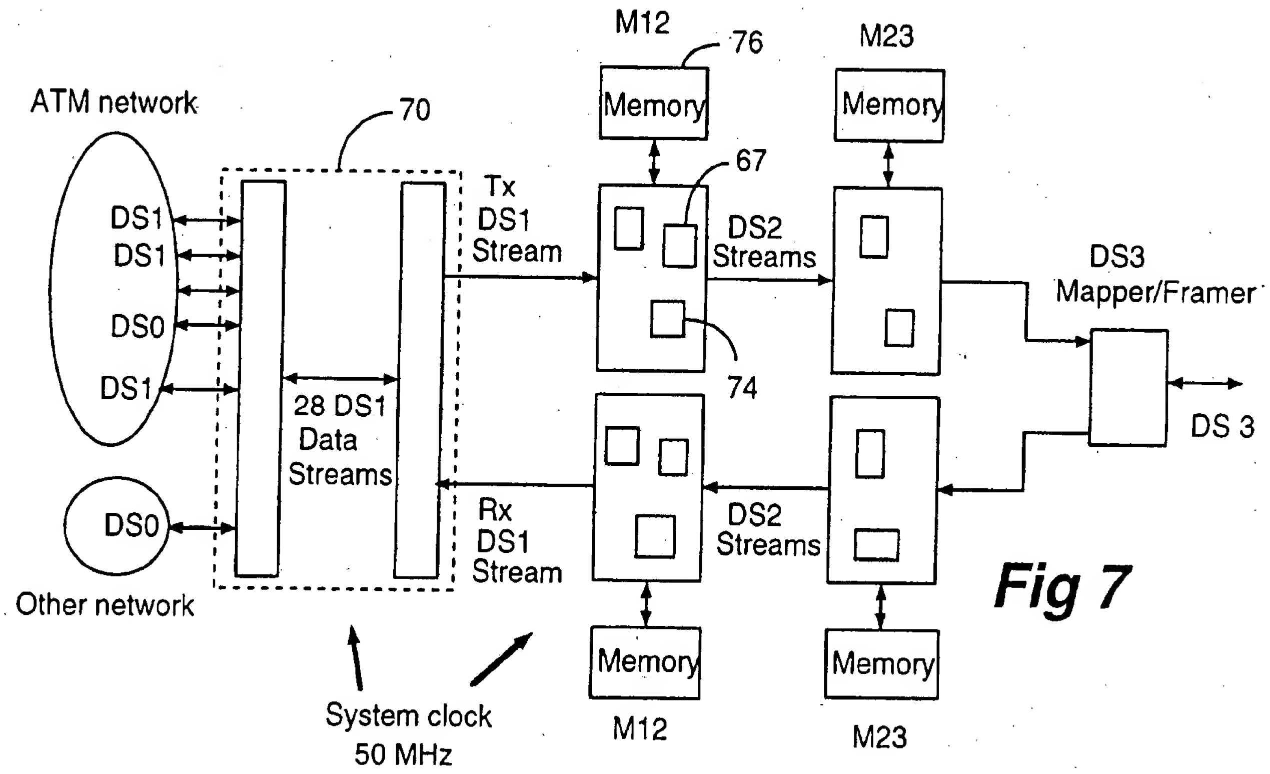

ATM network
DS1
DS1
DS0
DS1
DS0
Other network
70
28 DS1
Data
Streams
System clock
50 MHz
Tx
DS1
Stream
Rx
DS1
Stream
M12
76
Memory
67
74
DS2
Streams
M23
Memory
DS3
Mapper/Framer
DS 3
Memory
M12
DS2
Streams
Memory
M23

Fig 7

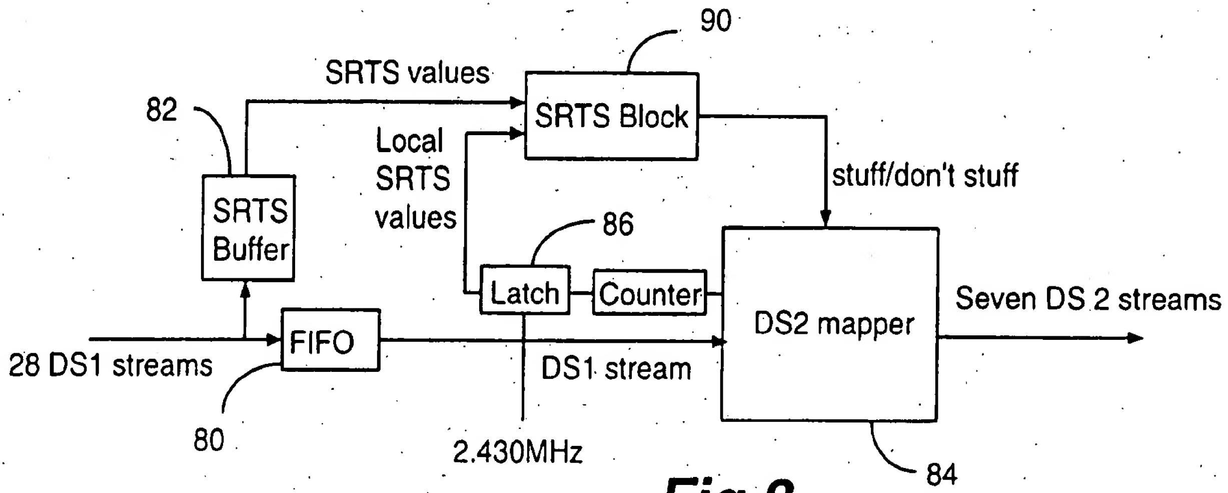

90
SRTS values
SRTS Block
82
Local
SRTS
values
stuff/don't stuff
SRTS
Buffer
86
Latch
Counter
DS2 mapper
Seven DS 2 streams
FIFO
28 DS1 streams
DS1 stream
80
2.430MHz
84

Fig 8

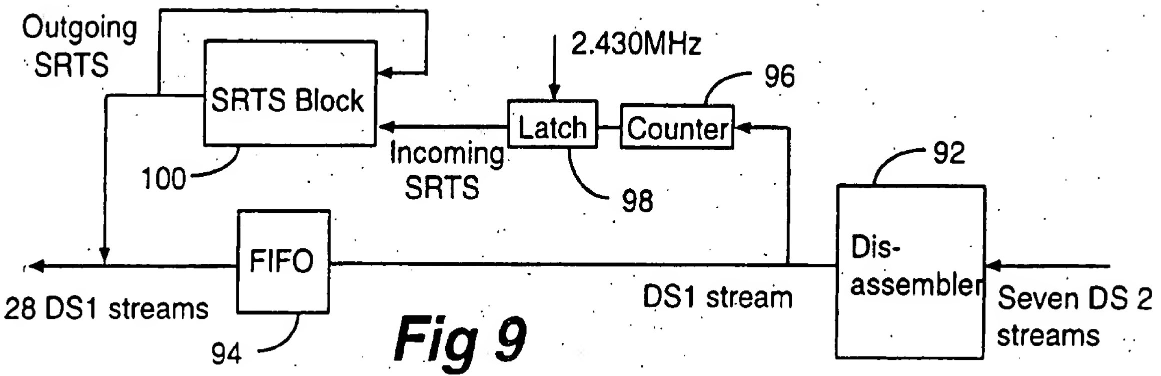

Outgoing
SRTS
2.430MHz
96
SRTS Block
Latch
Counter
Incoming
SRTS
92
100
98
Dis-
assembler
FIFO
28 DS1 streams
DS1 stream
Seven DS 2
streams
94

Fig 9

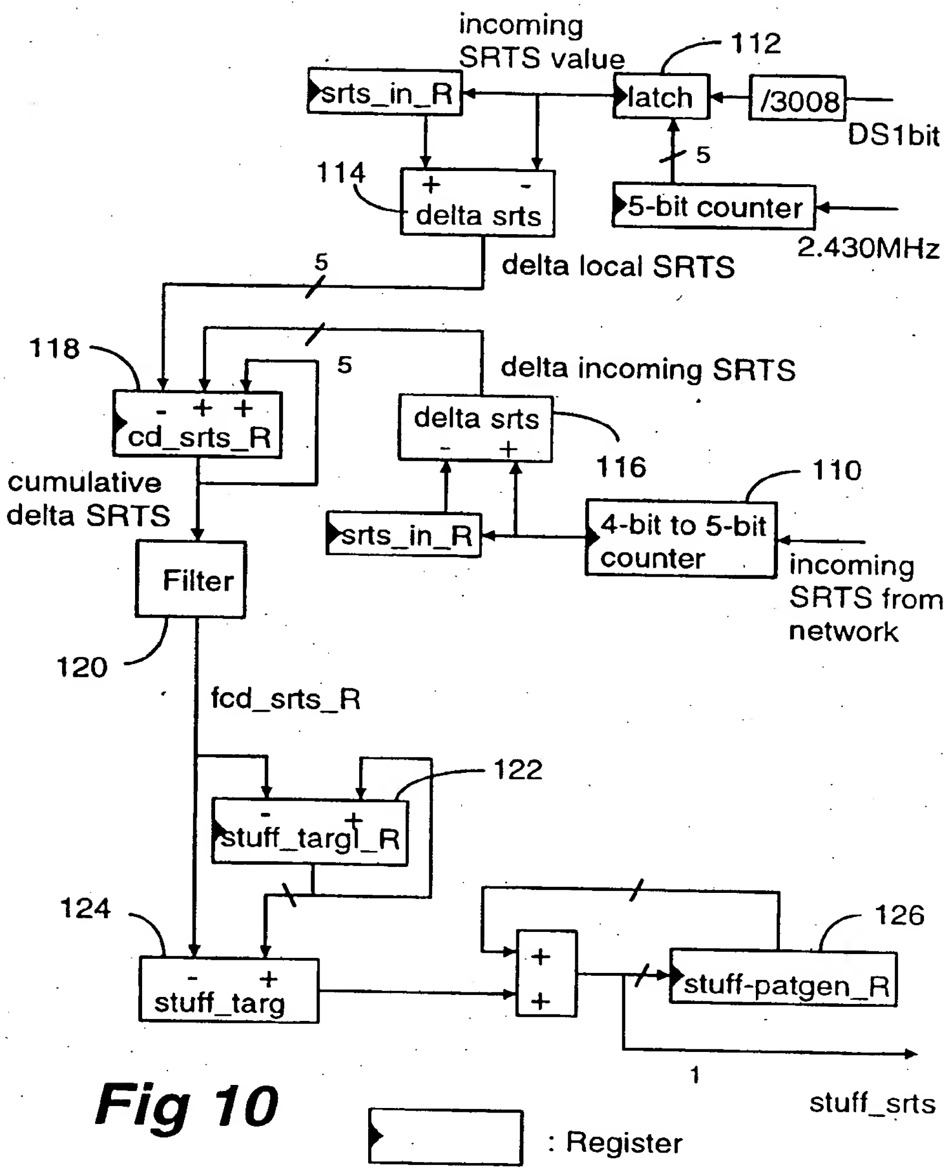

incoming
SRTS value
112
srts_in_R
latch
/3008
DS1bit
5
114
+
-
delta srts
5-bit counter
2.430MHz
5
delta local SRTS
118
5
delta incoming SRTS
- + +
cd_srts_R
delta srts
- +
116
110
cumulative
delta SRTS
srts_in_R
4-bit to 5-bit
counter
incoming
SRTS from
network
Filter
120
fcd_srts_R
122
- +
stuff_targl_R
124
126
- +
stuff_targ
+
+
stuff-patgen_R
1
stuff_srts
: Register

Fig 10

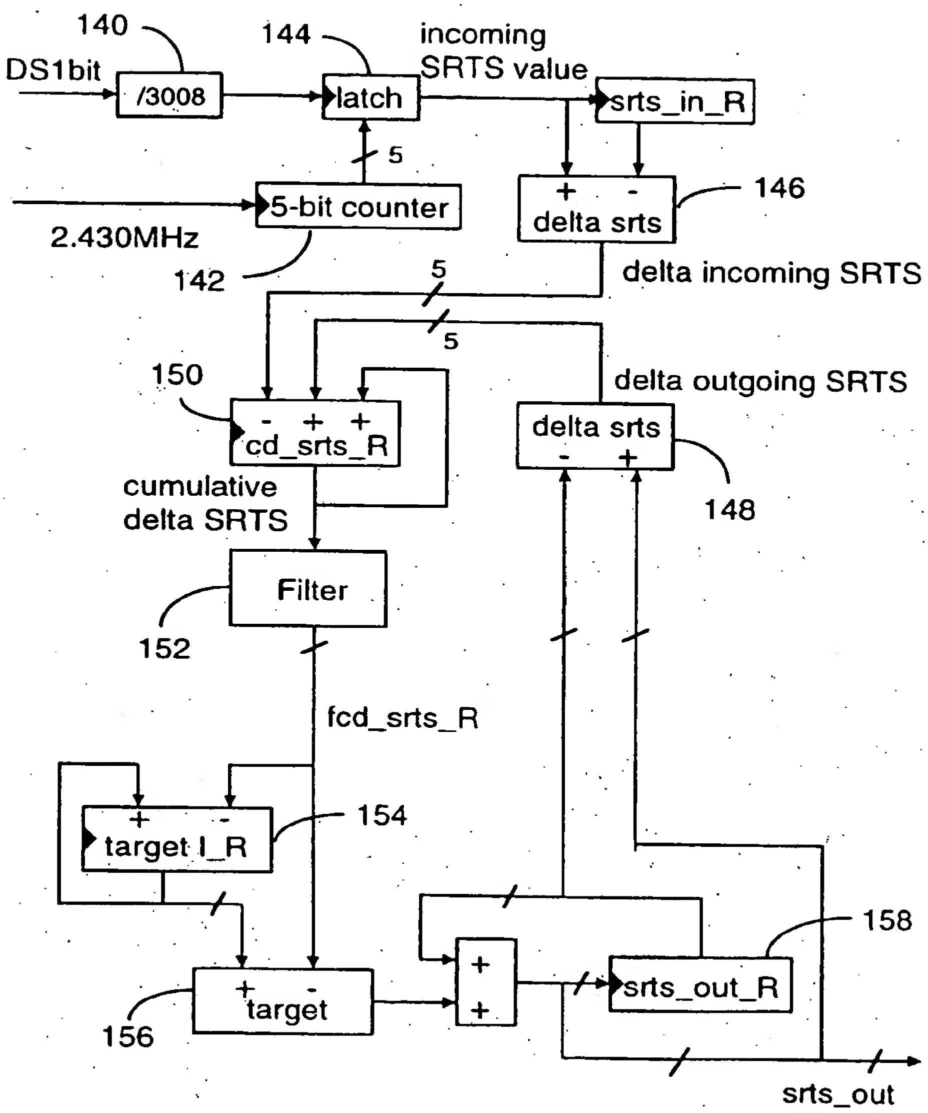

140
DS1bit
/3008
144
latch
incoming
SRTS value
srts_in_R
5
5-bit counter
2.430MHz
142
+
-
delta srts
146
5
delta incoming SRTS
5
150
- + +
cd_srts_R
delta outgoing SRTS
delta srts
- +
148
cumulative
delta SRTS
Filter
152
fcd_srts_R
+ -
target I_R
154
+ -
target
156
+
+
srts_out_R
158
srts_out

Fig 11

: Register